WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Wednesday, July 14, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR	
☐	L29	l16 and L25	8
☐	L28	l15 and L25	3
☐	L27	l12 and L25	3
☐	L26	l3 and L25	18
☐	L25	l20 or l21 or l22 or l23 or L24	3386
☐	L24	713/340.ccls.	443
☐	L23	713/300.ccls.	856
☐	L22	713/2.ccls.	848
☐	L21	713/1.ccls.	1006
☐	L20	710/5.ccls.	707
☐	L19	L17.clm.	1
☐	L18	L17.ab.	1
☐	L17	L16 same (based or depend$4 or respons$4)	21
☐	L16	(power$4 adj (down or up)) same (buffer$4 near3 (request or operation))	110
☐	L15	L14 same (determin$4 near5 (activ$6 or inactiv$6 or idle$4 or stall$4 or down))	74
☐	L14	(write near2 request)	10657
☐	L13	L12 same power$4	3
☐	L12	(file adj system) same (physical adj memory)	107
☐	L11	l2 same ((writ$4 or buffer$4 or read$4 or stor$4) with file with (portion or partial$4 or segment))	2
☐	L10	l3 same ((writ$4 or buffer$4 or read$4 or stor$4) with file with (portion or partial$4 or segment))	0
☐	L9	l3 same ((writ$4 or buffer$4 or read$4 or stor$4) near5 file near5 (portion or partial$4 or segment))	0
☐	L8	l3 same ((writ$4 or buffer$4 or read$4 or stor$4) near3 file near3 (portion or partial$4 or segment))	0
☐	L7	L3 same (buffer$4 with request)	0
☐	L6	L3 and (buffer$4 near5 request)	2
☐	L5	L3 same (buffer$4 near5 request)	0
☐	L4	L3 same (buffer$4 near3 request)	0
		(determin$4 near3 (power near2 (condition or state)) near3 (computer or	

☐	L3	device or unit))	213
☐	L2	(determin$4 near5 (power near2 (condition or state)))	1940
☐	L1	(regulat$4 near5 (file adj system) near5 access$4)	7

END OF SEARCH HISTORY

L1: Entry 1 of 7 File: USPT Feb 24, 2004

DOCUMENT-IDENTIFIER: US 6697944 B1
TITLE: Digital content distribution, transmission and protection system and method, and portable device for use therewith

CLAIMS:

44. A computerized portable device, comprising: storage media; a file system controlling the storage and retrieval of digital content files to and from said storage media; a USB interface; an authentication system regulating access to said storage media provided by said file system for transferring digital content files to and from said USB interface in accordance with a USB storage device class, said authentication system defining an authentication interface for said USB interface.

L1: Entry 2 of 7 File: USPT Sep 17, 2002

DOCUMENT-IDENTIFIER: US 6453334 B1
TITLE: Method and apparatus to allow remotely located computer programs and/or data to be accessed on a local computer in a secure, time-limited manner, with persistent caching

CLAIMS:

7. The file system driver program of claim 1, and including regulating access to the agent procedures themselves.

8. The file system driver program of claim 7, wherein the means to regulate access to the agent procedures includes at least one of: password checking procedures; execution counters; and timeout procedures.

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

☐ Generate Collection Print

L1: Entry 3 of 7 File: USPT Nov 23, 1999

DOCUMENT-IDENTIFIER: US 5991402 A
TITLE: Method and system of dynamic transformation of encrypted material

Brief Summary Text (28):
The present invention also operates in an network environment where access to material over a network file system is equally regulated and metered by the system.

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#
Generate Collection Print

L6: Entry 1 of 2 File: USPT Jun 5, 2001

DOCUMENT-IDENTIFIER: US 6243817 B1
TITLE: Device and method for dynamically reducing power consumption within input buffers of a bus interface unit

Abstract Text (1):
A computer is provided having a bus interface unit coupled between a CPU bus and a mezzanine bus, or PCI bus. The bus interface unit includes a plurality of input buffers which can be selectively connected and disconnected in a dynamic fashion according to active and inactive signals forwarded thereto. Signals forwarded to the bus interface unit from the CPU are classified according to the transaction phase of CPU bus activity. If signals associated with one particular transaction phase are active, then input buffers attributed to signals of other transaction phases can be deactivated. It is preferred that input buffers associated with signals of a request and arbitration phase be maintained active and coupled to power regardless of the present transaction phase unless the computer enters a powered down mode, such as sleep, idle or standby.

Brief Summary Text (16):
The present mechanism can also monitor the power state of the computer to determine if the computer is in a run state or a sleep state. If in a sleep state (including, e.g., idle state, standby state and hibernation state), then the CPU bus is monitored for a stop clock signal (STPCLK_) which indicates entry into the sleep state. If the stop clock signal is encountered, then all input buffers and/or differential amplifiers coupled to receive input signals from the CPU bus are disconnected from power. As defined herein below, power is the upper or lower power supplies, including V.sub.DD, V.sub.CC and/or ground.

Previous Doc Next Doc Go to Doc#

L6: Entry 2 of 2 File: USPT Feb 23, 1999

DOCUMENT-IDENTIFIER: US 5875120 A
TITLE: Information processing system

Detailed Description Text (5):
The data transfer performed between a CPU and an HDD is employed to briefly explain asynchronous communication and the handshaking operation (see FIG. 8). First, the CPU transmits a command (a write command in FIG. 8) to the HDD. Within about 20 msec following the transmission of the command, the HDD prepares for data buffering and then issues a data request. Commonly, a data transfer is performed by sectors (one sector is 512 bytes). Since I/O port address 1F7h that has a length of one word (=two bytes) is assigned to the data register for the HDD, 256 times access cycles are required for a single data transfer (i.e., 256 times sequential one word transfers). Each time one sector data transfer has been completed, the HDD issues an interrupt request (IRQ 14), and in response to the IRQ, the CPU (more specifically, the BIOS) accesses the status register for the HDD (I/O port address 1F7h) and confirms the status, i.e., the result of the write operation. The interrupt request and the status reading performed in response to the interrupt request constitute the so-called "handshaking operation". If the command from the CPU indicates data transfer for n sectors, the procedures for the data transfer and the handshaking operation are repeated n times.

Detailed Description Text (51):
FIG. 5 is a data writing operation that is asynchronously performed between the CPU 11 and the HDD 21. The data writing operation is divided into a command phase, a data buffering phase, a data request phase, a data transfer phase, a wait phase, and a handshaking phase.

CLAIMS:

4. The information processing system of claim 1, wherein said system is operative to perform a plurality of asynchronous data transfers cycles between said CPU and said peripheral device, and wherein said state determiner enters said second state and said power saving circuit directs said CPU to enter said power saving mode only in said asynchronous data transfer cycles that have a wait phase which is longer than an amount of time needed for said CPU to change from said power saving mode to said normal mode.

L11: Entry 1 of 2 File: USPT Mar 31, 1998

DOCUMENT-IDENTIFIER: US 5734894 A
TITLE: Methods and apparatus for protecting the integrity of process data stored on a removable storage medium

Detailed Description Text (73):
More particularly, with reference to FIG. 11, an exemplary set of process steps contemplated by the invention to perform the aforestated functions include: determining, after a power loss condition, if the non-volatile memory was previously configured (step 1101); determining if there are any data blocks that are marked as awaiting transfer to the removable medium (step 1102); determining if said medium is properly inserted in said instrument, i.e., inserted in the recording instrument with door 107 closed when using an instrument of the type depicted in FIG. 1 (step 1103); writing a block awaiting transfer to a target file on the medium to the recovery area of the target file, if the optional recovery area discussed hereinbefore (shown at 502 in FIG. 5) is used (step 1104); writing a block awaiting transfer to a target file on the medium, to the data storage portion (shown, for example, at 503-1 in FIG. 5) of the target file (step 1105); determining if the write operations performed in step 1104 (if performed) and step 1105 were successful (step 1106); marking each data block successfully transferred to the medium to indicate the transfer was complete whenever it is determined in step 1106 that the write operation(s) was (were) successful (step 1107); and determining if all blocks have been examined subsequent to experiencing a given loss of power condition (step 1108).

L11: Entry 2 of 2 File: USPT Dec 10, 1996

DOCUMENT-IDENTIFIER: US 5584029 A
TITLE: Data protecting system for an echangeable storage medium comprising power supply control means, medium detection means and medium identifying means

Brief Summary Text (29):
The present invention further comprises a first determination portion for determining whether or not the exchangeable storage medium inserted in the drive device is in an operable state as a file system in the information processing unit when a power supply stop request for the drive device takes place, a second determination portion for determining whether or not the exchangeable storage medium is in an accessible state when the result of the first determination portion is the operable state, and a power supply control portion for determining whether or not to stop a supply of electric power to the drive device according to the results of the first determination portion and the second determination portion, so as to control the supply of electric power to the drive device.

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L13: Entry 1 of 3 File: USPT Aug 27, 2002

DOCUMENT-IDENTIFIER: US 6442661 B1
TITLE: Self-tuning memory management for computer systems

Brief Summary Text (5):
However, conventional memory management methods, while enhancing performance in one network environment, often degrade performance in another. Even within one network environment, load changes from hour to hour can very widely and static memory management and tuning methods lead to degradation of the file server performance in many circumstances. Some conventional methods attempt to manage competition for memory by adding more physical memory or by using virtual memory methods, which rely on paging memory to buffer data to and from the storage system (e.g., disk drive). However, adding more physical memory is costly, requires space and consumes power. Further, virtual memory methods invariably lead to slow processing and unacceptable file server performance as they defeat the purpose of file system caching. Other conventional systems attempt to solve memory fragmentation by threading adjacent memory blocks together when freed. However, memory rethreading takes time, leading to reduced file server performance.

Previous Doc Next Doc Go to Doc#

L18: Entry 1 of 1 File: USPT Feb 21, 1989

DOCUMENT-IDENTIFIER: US 4807141 A
TITLE: Postage meter with microprocessor controlled reset inhibiting means

Abstract Text (1):
In a postage meter which includes a computer, a power supply for energizing the computer, a non-volatile memory for storing postage meter operating data, and wherein the computer includes a microprocessor adapted for processing the operating data, there is provided an improvement for protecting the operating data. The improvement comprises: the computer including (a) apparatus for detecting respective high level and low level output voltage signals from the power supply; (b) a first switching circuit operable in response to the detection of a high level output voltage signal for providing a power-up signal to the microprocessor and operable in response to the detection of a low level output voltage signal for providing a power-down signal to the microprocessor; (c) a second switching circuit operable in response to the detection of said high level signal for providing a not-reset signal to said microprocessor and operable in response to the detection of said low level signal for providing a reset signal to said microprocessor; and (d) apparatus for enabling operation of the non-volatile memory after the microprocessor has been provided with the power-up and not-reset signals, wherein the non-volatile memory enabling apparatus includes gate structure operable in response to timely receiving at least two respectively predetermined input signals for enabling the microprocessor to transfer said operating data between the microprocessor and the non-volatile memory, the non-volatile memory enabling apparatus includes a buffer circuit timely operable by the microprocessor for providing one of said two signals, and the microprocessor includes instrumentalities programmed for timely operating said buffer circuit and timely providing another of said two signals; and (e) third switching responsive to operation of the buffer circuit for inhibiting the operation of the second switching circuit to prevent the provision thereby of a reset signal to the microprocessor after the microprocessor is provided with a power-down signal and until the microprocessor has transferred the operating data from the microprocessor to the non-volatile memory.

L27: Entry 1 of 3 File: USPT Dec 2, 2003

DOCUMENT-IDENTIFIER: US 6658564 B1
TITLE: Reconfigurable programmable logic device computer system

Detailed Description Text (18):
Conventional operating systems such as Microsoft Windows NT schedule functions to be executed on the available resources in conventional computer systems at run-time. The resources in such a conventional computer system are allocated dynamically at run-time depending upon application program requirements and resource availability. Each resource has a resource manager, such as a virtual memory manager which is responsible for the allocation of physical memory. The most common resource managers in conventional operating systems are a file system manager, a graphics manager, an I/O manager (to handle mass storage access or simple I/O devices such as a keyboard or mouse), a network manager, and a virtual memory manager.

Current US Cross Reference Classification (2):
713/1

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L27: Entry 2 of 3 File: USPT Jan 5, 1999

DOCUMENT-IDENTIFIER: US 5857101 A
**** See image for Certificate of Correction ****
TITLE: Program lunch acceleration

Detailed Description Text (15):
Once the launch of a computer program is detected and a log file is opened, all file system activity is monitored (step 76). Specifically, for each operating system call to the file system the call is analyzed to determine to which application being launched, if any, does the call pertain. Exemplary operating system calls to the file system are OPEN, READ, WRITE, and CLOSE. If the call pertains to a computer program being launched, then an entry is appended to the appropriate log file (step 78). The log entry includes a file identifier, the logical address(es) specified in the call and the time of access (e.g., system time; index value). Alternatively, the physical memory address(es) corresponding to the logical address(es) are stored in the log entry. In some embodiments, the operating system has already caused the physical addresses to be generated. If not, then the physical addresses are derived from the logical addresses using the operating system's file allocation table to translate the logical address into the physical address.

Current US Original Classification (1):
713/1

Previous Doc Next Doc Go to Doc#

L27: Entry 3 of 3 File: USPT Mar 18, 1997

DOCUMENT-IDENTIFIER: US 5613123 A
**** See image for Certificate of Correction ****
TITLE: Method and system for configuring and executing device drivers based on configuration requirements

Detailed Description Text (28):
FIGS. 12A and 12B, respectively, illustrate a successful and unsuccessful configuration of a device driver based on satisfying and not satisfying the configuration requirements. In this example, sample configurations of various disk device drivers are shown. In each case, the device driver is configured to include a file system component driver, a disk caching component driver, a data compression component driver, an SCSI (Small Computer Systems Interface) component driver, and a bus interface component driver. The SCSI component driver performs device driver processing specific to the SCSI standard. The bus interface component driver performs device driver processing specific to a particular adapter, such as a SCSI adapter. The demand bits include an SCSI request bit, a physical memory address bit, and a physical disk address bit as the first three bit positions in the demand bit entry. Although these demand bits are used as examples, one of ordinary skill will readily understand that other demand bits can be provided which represent various requirements, such as logical or physical disk addressing, logical or physical memory addressing, alignment requirements, etc.

Detailed Description Text (36):
FIG. 12B shows a call-down table 1250 in which I/O access to the disk would not be allowed because not all the configuration requirements are satisfied. In the device driver represented by the call-down table 1250, all of the same component drivers are included as in FIG. 12A, except that the disk caching component driver does not provide physical memory address translation. As such, the bus interface component driver entry 1255, SCSI component driver entry 1254, and data compression component driver entry 1253 are linked into the call-down table and set and clear the same demand bits as the component driver entries 1215, 1214 and 1213 in the previous example. The disk caching component driver entry 1252, however, does not clear the physical memory address bit and thus stores "011" as the first three demand bits instead of the "001" stored in entry 1212. Thus, when the file system component driver clears the physical address bit in the topmost call-down table entry 1251, the physical memory address bit remains set, the first three demand bits being "010". Thus, when an I/O access to the disk is requested of the device driver configured in accordance with the call-down table 1250, the access will not be allowed because all bits in the topmost entry of the call-down table have not been cleared.

Current US Original Classification (1):
713/1

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L28: Entry 1 of 3 File: USPT Feb 18, 2003

DOCUMENT-IDENTIFIER: US 6523103 B2
TITLE: Disablement of a write filter stored on a write-protected partition

Detailed Description Text (22):
When the computer system is initially booted, the state machine retrieves 404, 406 its current state (i.e., RUNBOOT) and the active partition (i.e., PRIMARY). See FIGS. 4 and 8. Noting 408 that the current state is compatible with the active partition, the state machine then determines 414 that the write filter of the primary partition 204 is enabled in the partition's system registry (filter status=ON). Thus, the state machine causes the computer system 100 to boot to its primary partition 204. Once booted, the computer system 100 is free to execute the applications stored on the system's primary partition 204. Execution of some or all of these applications may be triggered by the state-specific code of the state machine. Due to the enabled write filter stored on the primary partition 204, all writes to the primary partition 204 are cached in RAM 200. However, the computer system 100 eventually receives a request to write persistent data to the primary partition 204, which request must be executed. Examples of requests which must be executed might include requests to update the computer system's IP (internet protocol) address, requests to upgrade the computer system's operating system or application components, etc. Write requests which must be executed may be identified as such by some form of tagging, or by application or state machine code which is able to identify certain pre-determined types of write requests. When such a request is received, either application or state machine code may cause the data associated with the write request to be written to persistent, write-enabled memory and/or mass storage. In FIG. 9, the data is written to a data partition 208 of the computer system 100 as persistent data. At some point, the state machine becomes aware of the need 500 to write persistent data to the primary partition 204, and the state-specific code 418' illustrated in FIG. 5 is executed. In response to the need to write persistent data to the primary partition 204, the state machine disables 502 the secondary write filter stored on the secondary partition 206. The secondary write filter may be disabled, for example, by renaming it to WRFILTER.BAK. The next state (PREPBOOT) and next active partition (SECONDARY) are then set 420, 422 and the computer system 100 is rebooted 412.

Current US Cross Reference Classification (3):
713/2

Previous Doc Next Doc Go to Doc#

L28: Entry 2 of 3 File: USPT Oct 23, 2001

DOCUMENT-IDENTIFIER: US 6308204 B1
TITLE: Method of communications for an intelligent digital audiovisual playback system

Detailed Description Text (65):
The task scheduler module will now be described in conjunction with FIG. 4. In the order of priority this module performs first test (761) to determine if the video task is active. In the case of a negative response it passes to the following test which is second test (751) to determine if the audio task is still active. In the case of a negative response third test (741) determines if the communications task is active. After a positive response to one of the tests, at stage (131) it fills memory access request queue (13) and at stage (132) executes this storage request by reading or writing in the mass storage, then loops back to the first test. When the test on communications activity is affirmative, scheduler (12) performs a test to determine if it is a matter of reading or writing data in the memory. If yes, the read or write request is placed in a queue at stage (131). In the opposite case, the scheduler determines at stage (743) if it is transmission or reception and in the case of transmission sends by stage (744) a block of data to the central server. In the case of reception the scheduler verifies that the kernel buffers are free for access and in the affirmative sends a message to the central server to accept reception of a data block at stage (747). After receiving a block, error control (748) of the cyclic redundancy check type (CRC) is executed and the block is rejected at stage (740) in case of error, or accepted in the opposite case at stage (749) by sending a corresponding message to the central server indicating that the block bearing a specific number is rejected or accepted, then loops back to the start tests. When there is no higher level task active, at stage (731 or 701) the scheduler processes interface or management tasks.

Current US Cross Reference Classification (4):
713/1

Current US Cross Reference Classification (6):
713/2

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

End of Result Set

Generate Collection Print

L28: Entry 3 of 3 File: USPT Mar 21, 2000

DOCUMENT-IDENTIFIER: US 6041366 A
TITLE: System and method for dynamic specification of input/output attributes

Detailed Description Text (14):
Referring now to FIG. 4, a method for dynamically specifying both a primary mirror and a scheduling policy for a write operation will now be described. The first step is to determine if the write request specifies a dynamic primary mirror (step 90). As discussed above, the dynamic primary mirror may be specified in a variety of ways. If no dynamic primary mirror is requested, the default setting established when the logical volume was defined is used (step 92). The default setting may either specify a primary mirror, or may indicate that the write request should be broadcast to all mirrors (i.e. parallel scheduling policy). If a dynamic primary mirror has been requested (i.e. the answer to the question in step 90 is "yes"), a check is made to determine if the requested dynamic primary mirror is active (step 94). If so, the device to be written to first is set to the requested dynamic primary mirror (step 96). If not, the default setting established when the logical volume was defined is used (step 98). As discussed above, the default setting may either specify a primary mirror, or may indicate that the write request should be broadcast to all mirrors (i.e. parallel scheduling policy).

Current US Original Classification (1):
710/5

Previous Doc Next Doc Go to Doc#

L29: Entry 1 of 8 File: USPT Dec 11, 2001

DOCUMENT-IDENTIFIER: US 6330679 B1
TITLE: Input buffer circuit with dual power down functions

Detailed Description Text (12):
In operation, input buffer circuit 32 includes two power down functions. The first power down function operates when laptop select input 66 is low (i.e., input buffer circuit 32 is not operating in a portable, battery operated computer). In this first function, when clock input 62 is idle, clock idle line 80 powers down all differential input amplifier buffers 50-53. Input buffer 60 remains powered up at all times to detect any activity of clock input 62. In this first power down function, clock input 62 controls the power down inputs of differential input buffers 50 and 51 through clock idle line 80 and controls the power down inputs of the remaining differential input buffers 52 and 53 through address strobe idle line 82.

Detailed Description Text (18):
In operation, input buffer 400 is powered up by applying a low signal to inverter 200. While powered up, the output of the resistor divider circuit inputs 1.0 volts to the gate of transistor 250 because the resistor divider circuit evenly splits the 2.0 volts to which it is coupled. Then, as the input signal at the gate of transistor 280 changes from ground to 2.0 volts and vice versa, the output of the current mirror at buffer 320 changes from ground to 3.3 volts and vice versa respectively.

Current US Original Classification (1):
713/300

L29: Entry 2 of 8 File: USPT Jun 5, 2001

DOCUMENT-IDENTIFIER: US 6243817 B1
TITLE: Device and method for dynamically reducing power consumption within input buffers of a bus interface unit

Abstract Text (1):
A computer is provided having a bus interface unit coupled between a CPU bus and a mezzanine bus, or PCI bus. The bus interface unit includes a plurality of input buffers which can be selectively connected and disconnected in a dynamic fashion according to active and inactive signals forwarded thereto. Signals forwarded to the bus interface unit from the CPU are classified according to the transaction phase of CPU bus activity. If signals associated with one particular transaction phase are active, then input buffers attributed to signals of other transaction phases can be deactivated. It is preferred that input buffers associated with signals of a request and arbitration phase be maintained active and coupled to power regardless of the present transaction phase unless the computer enters a powered down mode, such as sleep, idle or standby.

Current US Original Classification (1):
713/300

Current US Cross Reference Classification (5):
713/340

L29: Entry 6 of 8 File: USPT Aug 11, 1998

DOCUMENT-IDENTIFIER: US 5794032 A
TITLE: System for the identification and configuration of computer hardware peripherals

Detailed Description Text (25):
The system 100 determines the identity of the particular CD-ROM drive 30 and loads the corresponding software driver from the plurality of software driver files 104. This is in contrast to the conventional computer 10 (see FIG. 1) in which the user must identify the particular type of CD-ROM drive, locate the software driver file developed for that particular CD-ROM drive, and load the particular software driver file. When the conventional computer 10 is first powered up, or reset, the CONFIG.SYS file provides the file name of the correct software driver file for the particular CD-ROM drive 30 and sets other parameters, such as buffer size, interrupt request line, and the like. However, the system 100 includes software driver files 104 for all known types of CD-ROM drives 30 or all types of CD-ROM drives supported by the computer manufacturer. When the computer 10 is first powered up, or reset, the system 100 automatically identifies the particular type of CD-ROM drive 30 and loads the appropriate software driver file. Thus, the system 100 may be thought of having a specialized CONFIG.SYS file 106 that contains lines of code to load all the software driver files 104. The system 100 automatically identifies the CD-ROM drive 30 and modifies the specialized CONFIG.SYS file 106 to load only the appropriate software driver file from the floppy disk 102 without the need for the user to identify a particular type of CD-ROM drive 30. Similar procedures can be used with other peripheral devices. This is especially advantageous for less experienced users who may have difficulty identifying the particular model of the CD-ROM drive 30, or may have forgotten the relevant information if the installation was completed some time ago.

Current US Original Classification (1):
713/2

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L26: Entry 1 of 18 File: USPT Jul 6, 2004

DOCUMENT-IDENTIFIER: US 6760850 B1
TITLE: Method and apparatus executing power on self test code to enable a wakeup device for a computer system responsive to detecting an AC power source

Brief Summary Text (12):
Early implementations of the various power modes required the computer hardware itself to monitor user activity and determine the proper power state for each device in the computer system. These early computer systems included a read only memory (ROM) device that stored a set of instructions for the computer to follow in order to carry out power management functions. The set of instructions formed part of the Basic Input/Output System (BIOS) of the computer, which also included instructions for procedures such as accessing data on a hard or floppy disk drive and controlling the graphics display. The ROM device containing the BIOS is referred to as the "BIOS ROM". Because hardware-based power management instructions usually are included in BIOS, such a management scheme is commonly known as "BIOS power management". Under BIOS power management, conditions within the computer system that initiate power state transitions, such as button presses and periods of inactivity explained above, generate system management interrupt (SMI) signals to the central processing unit (CPU). Upon receiving an SMI, the CPU executes the BIOS power management instructions stored in ROM to change the power state.

Current US Cross Reference Classification (1):
713/300

Current US Cross Reference Classification (6):
713/340

Previous Doc Next Doc Go to Doc#

L26: Entry 2 of 18 File: USPT Apr 13, 2004

DOCUMENT-IDENTIFIER: US 6721885 B1
TITLE: Reducing start-up time and avoiding customer-induced system failures for personal computers

Current US Original Classification (1):
713/2

CLAIMS:

10. The method of claim 9, further comprising the steps of: determining when said computer system is in a power suspended state; and in response to determining that said computer system is in said power suspended state, loading an operating system during said power-up if said signal has been received.

12. A method for reducing configuration-based crashes of a computer system, said method comprising the steps of: determining during power suspended state if a system unit of said computer system has been opened; and in response to said determining step and receipt of a request for power-up of said computer system, initiating a full power on self test (POST) operation when said system unit has been opened and returning said computer system from said power suspended state without said POST operation when said system unit has not been opened.

14. A method for reducing power-up time of a computer system, said method comprising the steps of: determining during power down state if a system unit of said computer system has been opened, wherein said power down state removes power from each major system component and said determining step is completed by a switch circuit connected to a cover portion of said computer system and having an independent power source from said system components; and in response to said determining step and a request for power-up of said computer system, initiating a full power on self test (POST) operation when said system unit has been opened and powering up said computer system without completing said POST operation when said cover has remained closed, wherein a computer system powers up from a complete power down state, as well as a suspended state without completing a POST operation and wherein further said computer system completes a complete POST operation when the system has been opened during an initial power-up or suspend state prior to power down.

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#
Generate Collection Print

L26: Entry 11 of 18 File: USPT Apr 18, 2000

DOCUMENT-IDENTIFIER: US 6052793 A
TITLE: Wakeup event restoration after power loss

Brief Summary Text (16):
Accordingly, one aspect of the present invention provides a method of re-enabling a computer system wakeup event after a computer system power on reset Whether the power on reset was caused by an invalid event is determined. When the power on reset was caused by an invalid event, wakeup event information is read from a nonvolatile memory, a wakeup event register is configured so as to enable the wakeup event, and the computer system is placed in a power state. The power state is determined by the power state of the computer system prior to the invalid event

Detailed Description Text (6):
If the reset is due to an invalid event, the BIOS then determines if the pre-reset power state was the fully on power state, as shown in step 240. Information in nonvolatile memory, such as a power state flag, is examined by the BIOS in order to determine the appropriate power state for the computer system. If the power state flag indicates that the computer system was not in the fully on state prior to the invalid event, the appropriate wakeup event register or registers are restored as indicated in 250. For example, if the wakeup on LAN wakeup event was previously enabled, this information will be indicated in the nonvolatile memory. Using this information, the BIOS resets the necessary bit or bits in the power management circuit's register or registers. The BIOS then returns the system to the soft-off state, 275, without having to continue execution of the BIOS code.

Current US Original Classification (1):
713/340

CLAIMS:

1. A method of re-enabling a computer system wakeup event after a computer system power on reset, the method comprising:

determining whether the power on reset was caused by an invalid event; and

when the power on reset was caused by an invalid event:

reading wakeup event information from a nonvolatile memory;

configuring a wakeup event register so as to enable the wakeup event; and

placing the computer system in a power state, the power state being determined by the power state of the computer system prior to the invalid event.

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#

Generate Collection Print

L26: Entry 12 of 18 File: USPT Apr 11, 2000

DOCUMENT-IDENTIFIER: US 6049879 A

**** See image for Certificate of Correction ****

TITLE: Power management diagnostic in electronic devices

Brief Summary Text (10):

To meet the requirements of the energy conservation programs, a computer must include a reliable power management system. However, power management decisions may be made by applications, device drivers, BIOS and the OS. Some decisions of an application, for example, may conflict with decisions made by another. Identification and debugging of such conflicts cannot be made without a diagnostic to monitor the power states of the devices and to determine when devices change power states and to determine which of the applications, device drivers, BIOS or OS cause the device to change power states.

Detailed Description Text (21):

The master terminal 100 commands the target terminal 200 to enter a monitoring mode (Steps 4030 and 4035). In this mode, the target terminal 200 monitors applications and device drivers to determine when they cause changes in the power states of any device. When a power change occurs in any device within the target terminal 200, it is an "event."

Current US Original Classification (1):

713/300

Previous Doc Next Doc Go to Doc#